IN THE CLAIMS:

1. (Currently Amended) A method for reducing latencies associated with accessing memory for more than one processors, each coupled with an associated private cache, the method comprising:

determining cache miss rates of the more than one processors when issuing cache requests against one or more private caches;

comparing the cache miss rates of the more than one processors; and

allocating cache lines from a first private cache associated with a first processor to a second processor based upon the difference between the cache miss rate for the first processor and the cache miss rate of the second processor, wherein a latency to access the allocated lines of the first private cache by the second processor is greater than a latency to access cache lines of a second private cache associated with the second processor, and wherein the first private cache and the second private cache are at a same cache level.

2. (Original) The method of claim 1, wherein determining the cache miss rates comprises counting cache misses of each of the more than one processors.

3. (Previously Presented) The method of claim 1, wherein allocating cache lines comprises forwarding cache requests from the first processor to the second private cache associated with the second processor.

4. (Previously Presented) The method of claim 1, wherein allocating cache lines comprises selectively allocating cache lines based upon a priority associated with a cache request of the first processor.

5. (Currently Amended) A method for reducing cache miss rates for more than one processors, wherein the more than one processors couple with private caches, the method comprising:

monitoring the cache miss rates of the more than one processors;

comparing the cache miss rates of the more than one processors to determine when a cache miss rate of a first processor associated with a first private cache of the private caches exceeds a threshold cache miss rate for the more than one processors;

forwarding a cache request associated with the first processor to a second private cache of the private caches in response to determining the cache miss rate exceeds the threshold cache miss rate;

replacing a cache line in the second private cache with a memory line received in response to the cache request; and

accessing the cache line in response to an instruction from the first processor, wherein a latency to access the cache line of the second private cache by the first processor is greater than a latency to access cache lines of the first private cache associated with the first processor, and wherein the first private cache and the second private cache are at a same cache level.

6. (Previously Presented) The method of claim 5, wherein monitoring the cache miss rates comprises not counting cache misses during a cold start, warm-up period.

7. (Previously Presented) The method of claim 5, wherein the cache miss rates are associated with more than one processor modules.

8. (Original) The method of claim 5, wherein the threshold cache miss rate is based upon an average cache miss rate for the more than one processors.

9. (Original) The method of claim 5, wherein forwarding the cache request comprises selecting the second private cache based upon a least recently used cache line associated with the private caches.

10. (Original) The method of claim 9, wherein selecting the second private cache comprises selecting a least recently used cache line based upon a processor module on which the first processor resides.

11. (Original) The method of claim 5, wherein forwarding the cache request comprises selecting the cache request based upon a priority associated with the cache request.

12. (Original) The method of claim 5, wherein forwarding the cache request is responsive to a software instruction that overrides a result of comparing the cache miss rates to forward the cache request to the second private cache.

13-42. (Cancelled)